

A Modeling and Controls Course using Microcontrollers

Hugh Jack, Associate Professor
School of Engineering, Grand Valley State University

Abstract

Microcontrollers are being used to support the laboratory and project components of a junior level mechanical/manufacturing engineering course in system modeling and control. Previously the course primarily used Labview and Data Acquisition (DAQ) cards with Personal Computers. However, upon completion of the course students were not able to design and implement practical control systems. Recently the course has been evolving towards low cost commodity hardware found in mass produced consumer products. This past year the course used custom designed circuit boards based on the Atmel Mega32 microcontroller. While these new boards are low cost (\$37), they support a much more relevant and mature set of pedagogical experiences. The objective of the paper is to present enough information to allow other educators to evaluate the viability of the controllers for use in their own courses. All of the board designs, lab descriptions, sample programs and course textbook are available on the course home page [1].

Introduction

EGR 345 - Dynamic System Modeling and Control is a course offered to junior level mechanical and manufacturing engineering students [1]. The course topics, listed below, include the modeling, analysis, and control of mechanical and electrical systems starting with differential equations. Prerequisites for the course include EGR 261 - An introduction to C programming, EGR 226 - An Introduction to Digital Systems, EGR 214 - Circuit Analysis I, EGR 209 - Statics and Mechanics of Materials, and EGR 101 - CAD/CAM.

- Translation
- Solving Differential Equations
- Numerical Methods
- Rotation
- Transfer Functions
- Circuits
- Feedback Control Systems and Block Diagrams
- Phasor Analysis
- Bode Plots
- Root Locus Plots
- Non-linear Systems
- Analog IO
- Sensors
- Actuators
- Motion Control Systems
- Laplace Techniques

The lectures are supported by weekly laboratory exercises and a semester long project. Until 2002 the laboratory exercises were primarily based on Labview software and Data Acquisition Cards for measurement and control. This approach was convenient for rapidly prototyping simple sys-

tems, however it hid many of the important system details. The result was that students had chronic problems with issues such as data types and timing. In addition, the high cost of these systems limits their use to a narrow set of industrial applications. In response to these issues a new laboratory strategy was developed using low cost microcontrollers and C programming. In the fall of 2003 the laboratories were revised to use commercial (Axiom) 68HC11 microcontrollers boards for a majority of the data acquisition and control tasks [2]. Although based on decade old technology, these boards were flexible, low cost (\$89) and relatively reliable. In the fall of 2004 the laboratory was revised to use custom designed boards based on the Atmel Mega32 microcontroller. These boards have been designed to be very reliable and low cost (\$37). In the fall of 2004 the parts for 20 boards were purchased by the department and boards were assigned to laboratory groups of three students for the semester. It is worth noting that the total cost for the boards was less than the cost for a single DAQ card.

The Board Design

The design of the new microcontroller board was based upon our experiences with the Axiom 68HC11 boards in the fall of 2003. In general students were comfortable with the 68HC11 platform and could program it in C using the GCC compiler. In most circumstances these 68HC11 boards worked well but there were issues of availability (we had 8). Later in the semester when the students began driving motors with the boards there were numerous failures of the 32K static RAM IC and the RS-232 driver IC. These problems were attributed to a lack of noise isolation and protection. There were also problems with students damaging boards by carelessly applying voltages to the wrong pins. In some cases a student would damage multiple boards (using trial and error debugging) before the problems were discovered. A couple of times during the semester we would find that more than half of the boards had been damaged. In addition, the 68HC11 microcontroller lacked a few features that were very desirable including multiple Pulse Width Modulated (PWM) outputs and nonvolatile memory.

The new microcontroller board design, shown in Figures 1 and 2, was designed to overcome many of these problems. The 68HC11 microcontroller was replaced with the Atmel Mega32 microcontroller. It has 32K of flash (nonvolatile) memory and four PWM outputs. For the laboratories the programming is done with C using function calls similar to those used with the 68HC11. The 40 pin DIP Mega32 package was selected so that students with limited soldering experience could reliably mount it using a socket. All of the critical ICs are mounted using sockets to minimize heating damage during soldering and to be easily replaced.

The board was designed using Eagle, which is free for academic uses, however there is a maximum board size of 3 inches by 4 inches. This had the unexpected, but appreciated benefit of forcing the board into a less bulky form. To minimize the damaging effects from electrical noise and erroneous connections, current limiting resistors, bypass capacitors, fuses, and zener diodes were added. Labels were used to minimize the confusion when making connections. Terminal strips were added to accommodate stranded conductors and reduce loose wiring.

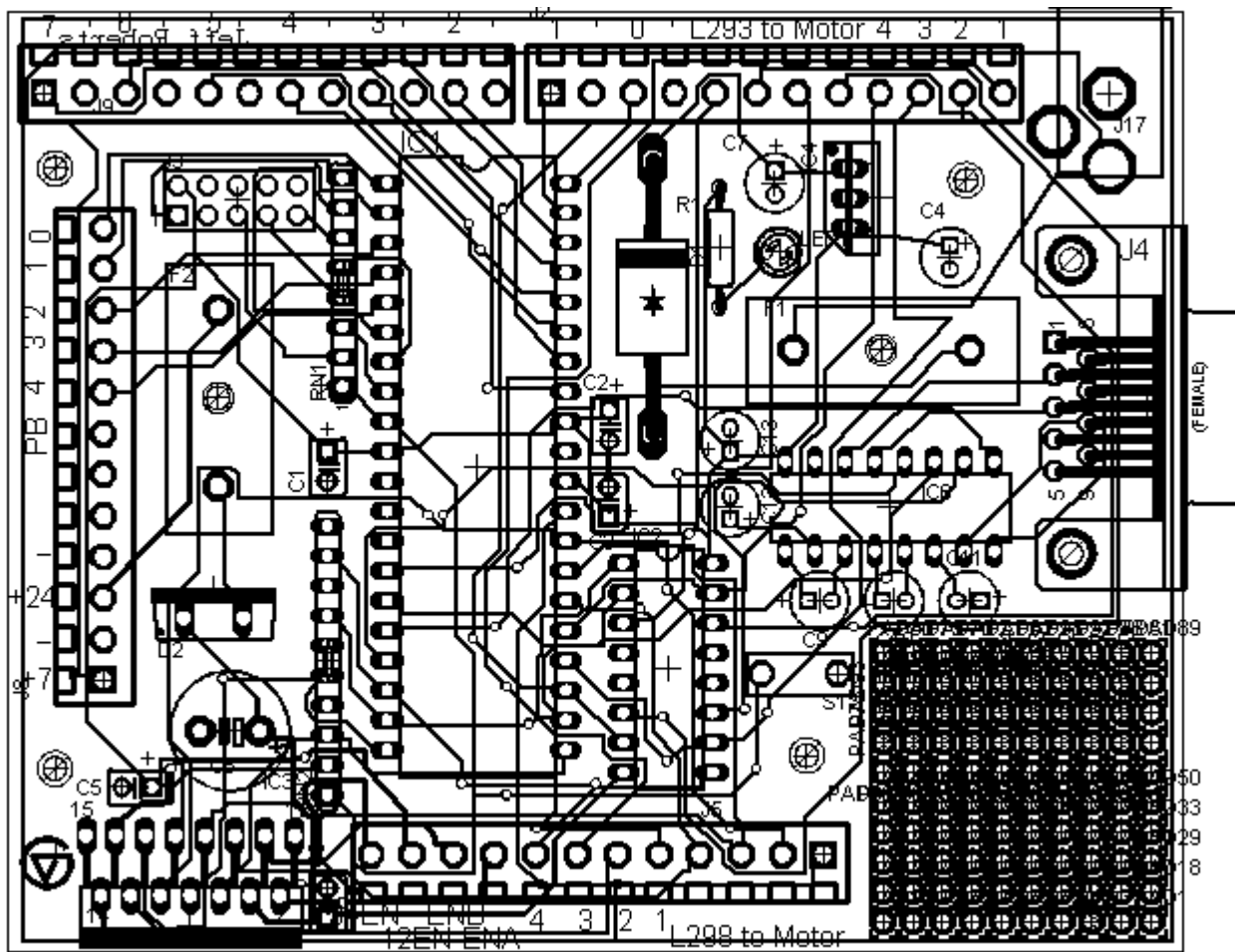


Figure 1 - PCB Layout (the Eagle source file is available on the course web page [1])

To increase student accountability, and discourage trial-and-error debugging, the boards were assigned to groups of students for the semester. If a board was damaged the group needed to spend the time and effort to make the necessary repairs. This year there were substantially fewer problems - only 3 out of 16 groups encountered hardware problems during the semester. These groups required some intervention by the instructors who would then diagnose what the group had done to cause the problem.

Some of the problems listed below were encountered in the fall of 2004 but will be fixed in the next iteration of the design to be used in the fall of 2005.

- Problems occurred when trying to download programs using the RS-232 interface with laptops that only had USB ports. USB to serial converters would cause problems because of timing latencies in the serial interface. The serial port will be replaced with a USB port.
- ATmega32 ICs were 'blown' by students applying analog voltages greater than the 5V limit. Overvoltage protection will be added for the first couple of analog input channels.
- The L293 H-bridge IC did not supply enough current for some motors, it will be replaced with an L298.

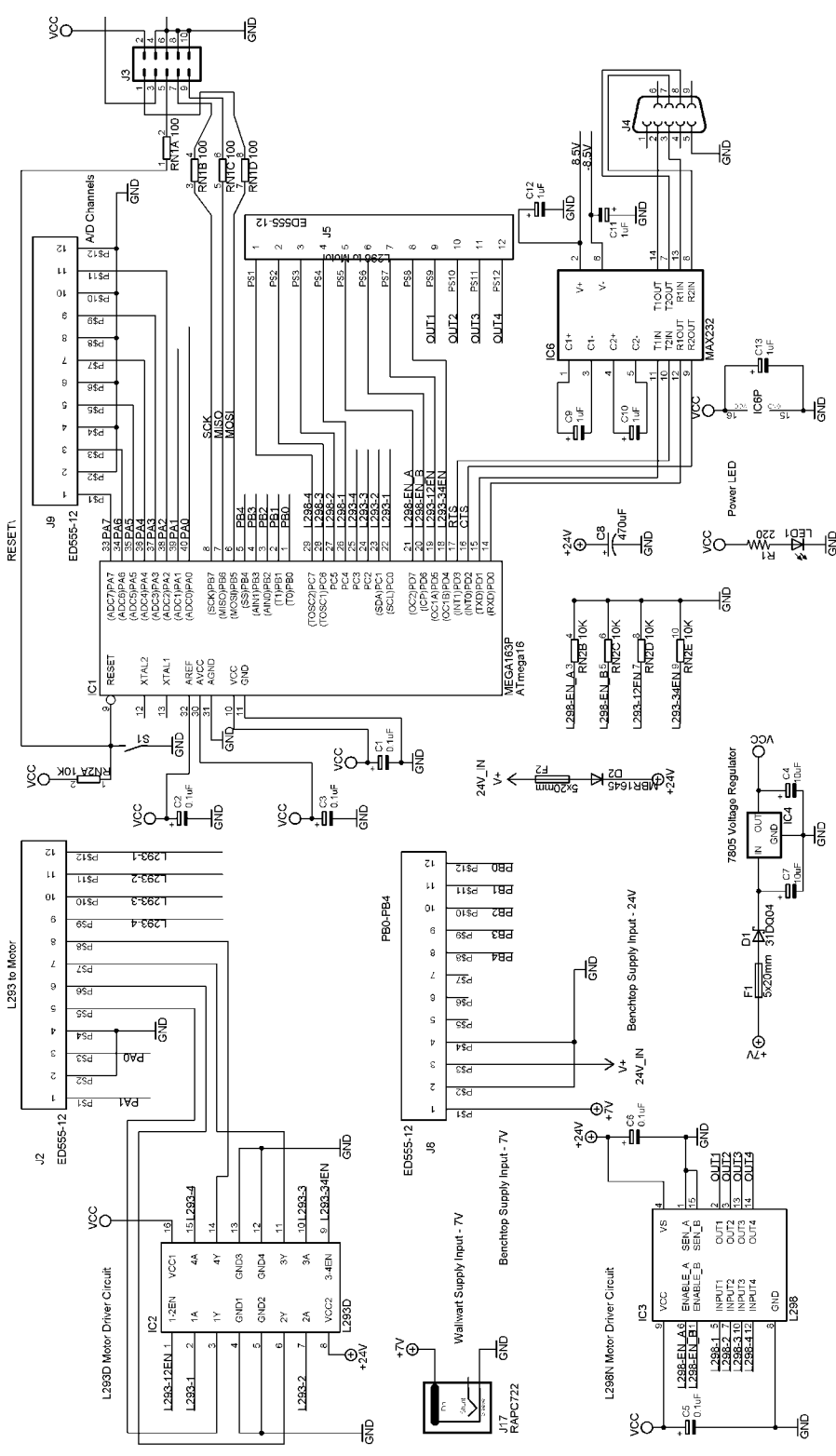


Figure 2 - Atmel Mega 32 Controller Schematic (the Eagle source file is available on the course web page [1])

The Laboratory Exercises

The laboratory sequence started with students building the boards. They then learned to program the Mega32 in C and use it for analog inputs, interrupt driven timing, motor feedback control, data acquisition, dead-band compensation, transistor, and H-bridge control via pulse width modulation and motion profiles. The weekly exercises and brief descriptions are listed below. The detailed laboratory descriptions are available on the course web page [1].

1. Atmel Board Assembly - The ATMega 32 PCBs were assembled and tested using a step-by-step guide.
2. Programming in C - A basic programming tutorial on using the ATMega 32 for various I/O and interrupt tasks.
3. Numerical methods - Scilab and C programs were used for numerical solutions.
4. Feedback Control - A proportional feedback control system using a PWM controlled transistor and tachometer for velocity feedback.
5. Dead-band Compensation - Stiction values were measured and then used in compensation subroutines.
6. Position Control - An H-bridge was used to allow bidirectional motion with a potentiometer for position feedback.
7. Motion Control - Subroutines were used to generate setpoints for motions that would start and stop smoothly.
8. DC Motor Modeling - Parameters for an ideal DC motor were measured/calculated.
9. System Modeling - The motor model was used to compare theoretical and actual responses of a feedback control system.
10. Variable Frequency Drives - Industrial Variable Frequency (VFD) drives were used with AC motors.
11. Labview - Industrial software and hardware was used for data input and output.
12. Simulink - The block diagram simulation tool in Matlab was used for system modeling.

With the new focus on microcontrollers the students were able to implement a greater number of more advanced control systems during the course. Moreover, when they use Labview for the first time in laboratory 11 they write more advanced programs than students were able to in previous years, even when they had taken the Labview only version of the course. The knowledge students obtained in the laboratory was applied in the semester project.

The Project

The semester project is formally organized to have teams of students solve a complicated design problem. In the fall of 2004 the project was a two wheeled self balancing robot that could follow a line. During these design projects the students designed all aspects of the device following given objectives and constraints including cost, weight, and performance. They then built and tested their robots. A few of the 11 robots built are shown in Figure 3.

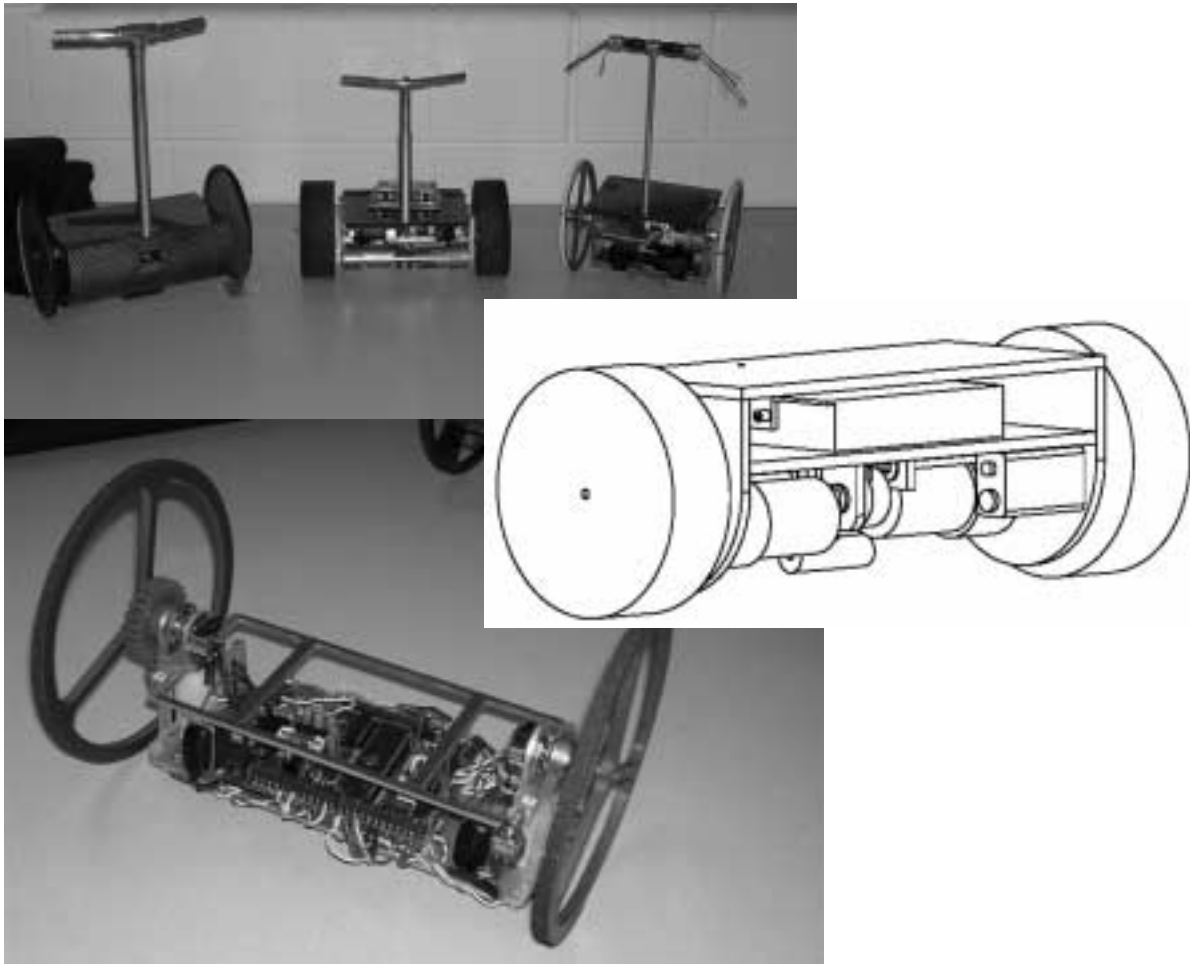


Figure 3 - Self Balancing Robots.

The design objectives and constraints were selected so that the students could use the microcontroller boards and apply the knowledge obtained during the laboratories. For this design the students selected a variety of sensors for tilt sensing including hanging masses with potentiometers, optical distance sensors and accelerometers. To sense the lines students were provided with CdS photocells. The students were encouraged to keep the costs low (the actual range was \$114 to \$145) and the weight low (the actual range was 500g to 1.2Kg). The results of these efforts can be seen on the course home page [1].

Conclusion

The results of using microcontrollers continues to be highly successful. On a weekly basis in the laboratory, students implement complex control systems using C programs. Next year the boards will be simplified by reducing the terminal strips to reduce the cost. The target price for the boards will be less than \$25. With the price in this range the boards will become suitable for general purchase along with other course supplies.

Acknowledgements

This work was made possible with the help of others. Jeffry Roberts developed the final board and tutorials based upon work begun by Nancy Kindraka. Dr. Andrew Sterian provided valuable assistance with the Atmel architecture. Dr. Andy Blauch helped work our many of the bugs and added some software tools in the laboratory.

References / Bibliography

- [1] Jack, H., "Dynamic System Modeling and Control" course homepage, <http://claymore.engineer.gvsu.edu/eod/egr345.html>
- [2] Jack, H., Blauch, A., "A Modeling and Controls Course Using Microcontrollers", ASEE Annual Meeting, Salt Lake City, June 2004.
- [3] Murray, W. R., Garbini, J. L., "Embedded Computing in the Mechanical Engineering Curriculum: A Course Featuring Structured Laboratory Exercises", *Journal of Engineering Education*, pp. 285-290 July, 1997.
- [4] Moallem, M., "Design and Implementation of Computer Control Software", *IEEE Control Systems Magazine*, pp. 26-29, February 2005.
- [5] Arzen, K-E., Blomdell, A., Wittenmark, B., "Laboratories and Real-Time Computing", *IEEE Control Systems Magazine*, pp. 30-34, February 2005.

Biography

HUGH JACK earned his bachelors degree in electrical engineering, and masters and Ph.D. degrees in mechanical engineering at the University of Western Ontario. He is currently an associate professor at Grand Valley State University and chairs the graduate and manufacturing programs. His research interests include controls and automation, including the use open source software for industrial control.